Methods and Tools for Interpretable Bayesian Variable Selection

Markus Paasiniemi

School of Science

Thesis submitted for examination for the degree of Master of Science in Technology.

June 17, 2018

Thesis supervisor and advisor:

Prof. Aki Vehtari

Aalto University
School of Science

AALTO UNIVERSITY
SCHOOL OF SCIENCE

ABSTRACT OF THE
MASTER'S THESIS

Author: Markus Paasiniemi

Title: Methods and Tools for Interpretable Bayesian Variable Selection

Date: June 17, 2018 Language: English Number of pages: 4+53

Department of Computer Science

Major: Machine Learning and Data Mining

Supervisor and advisor: Prof. Aki Vehtari

This thesis discusses interpretability in model selection. It considers some of the central themes of interpretable models and introduces a new tool, shinyproj, to improve interpretability in variable selection. shinyproj is a new R package for interpretable Bayesian model selection for generalised linear models. shinyproj emphasises a modern workflow for variable selection, in which the properties of the models are examined iteratively with a guidance of an efficient variable selection algorithm. The need for the package is motivated especially by the increasing demands for transparent and interpretable models, which are also discussed in this thesis. The problem is that in order to increase the performance of the model, one often has to increase the complexity of the model, which in turn will often reduce the interpretability of the model. shinyproj combines an existing R package for projection predictive variable selection with an interface that allows the user to explore the model space and make informed and efficient tradeoffs between the accuracy and the interpretability of the model. While the current functionality of the package does not constitute a conclusive solution to the problem, it serves as a proof-of-concept and likely a good basis for future improvements.

Keywords: Bayesian model selection, interpretable models, variable selection, forward selection, projection

AALTO-YLIOPISTO
PERUSTIETEIDEN KORKEAKOULU

DIPLOMITYÖN
TIIVISTELMÄ

Tekijä: Markus Paasiniemi

Työn nimi: Tulkittavia menetelmiä ja työkaluja bayesiläiseen mallinvalintaan

Päivämäärä: June 17, 2018 Kieli: Englanti Sivumäärä: 4+53

Department of Computer Science

Professuuri: Machine Learning and Data Mining

Työn valvoja ja ohjaaja: Prof. Aki Vehtari

Tämä työ tarkastelee tulkittavaa bayesiläistä mallinvalintaa. Yhtäältä työssä tarkastellaan tekijöitä, jotka tekevät malleista tulkittavia, mutta toisaalta työssä esitetään myös uusi työkalu, shinyproj, joka tekee lisäksi itse mallinvalintaprosessista ymmärrettävän. shinyproj on uusi R paketti tulkittavaan bayesiläiseen mallinvalintaan yleisestetyille lineaarimalleille (generalized linear models). shinyproj korostaa moderneja työskentelytapoja, jossa mallin ominaisuuksia tarkastellaan iteratiivisesti tehokkaan muuttujanvalinta-algoritmin tuella. Tulkittavien ja ymmärrettävien mallien tarve on noussut erityisesti viime aikoina, kun yhtäältä malleja käytetään enemmän ja enemmän osana päätöksentekoa, mutta toisaalta juuri siitä syystä malleilta vaaditaan myös läpinäkyvyyttä ja tulkittavuutta. Ongelmana on pohjimmiltaan se, että mitä paremmin mallin halutaan suoriutuvan, sitä monipuolisempi ja yksityiskohtaisempi sen on vääjäämättä oletava. Monipuolisuus ja yksityiskohtaisuus taas tekevät mallista väkisinkin vaikeammin tulkittavan ja ymmärrettävän. shinyproj yhdistää olemassaolevan tehokkaan parametrien projisoimiseen perustuvan muuttujanvalinta-paketin yksinkertaiseen graafiseen käyttöliittymään, joka helpottaa malli-avaruuden läpikäymistä ja siten mahdollistaa informoitujen ja tehokkaiden vaihtokauppojen tekemisen mallin suorituskyvyn ja tulkittavuuden välillä. Vaikka nykyisellään paketti ei ratkaisekkaan tyhjentävästi kaikkia tulkittavaan mallinvalintaan liittyviä ongelmia, se tarjoaa siihen yhden käyttökelpoisen ratkaisun ja toimii esimerkkinä siitä, minkälaisia ratkaisuja ongelmaan voi tulevaisuudessa tarjota.

Avainsanat: Bayesiläinen mallinvalinta, tulkittavat mallit, muuttujanvalinta

Contents

1 Introduction

One of the most essential questions related to modelling is model selection. Model selection means the process of choosing a suitable model for the task at hand, based both on the model's function in said task as well as its characteristics. Since for most modelling problems it is not clear whether a 'true' model even exists, the commonly adopted strategy is to try to find models that are the most useful. A typical metric of usefulness is that if a model does not yield predictions that match the reality at all, there is likely little to be learned from it. Conversely, when the predictions made by the model are accurate and match reality well, being able to interpret the mechanisms identified by the model from the data is likely to be useful.

In addition, it is becoming increasingly important that the selected models are also understandable to the users of the models as modelling is adopted more and more to aid important real-life decision-making, such as in health care, finance or the criminal justice system. The requirement of interpretability will also become even more concrete in the EU in 2018 as the EU General Data Protection Regulation will come in to force. Among other things, the regulations effectively create a 'right for meaningful explanation' for citizens that are subject to these models (Goodman and Flaxman, 2016), posing demands for the models used in automated decision-making.

This thesis considers the model selection for Bayesian generalised linear models especially with interpretability in mind. The problem is that while increasing complexity of a model will tend to increase its predictive power, it will likely also lead to increased difficulty in interpreting the model. A major issue is that at the moment, there exist methods that help construct accurate models, with little concern for interpretability. Even though predictions may be accurate, this means that in some ways, the usability of the model suffers. On the other hand, there also exist tools specifically for building interpretable models, but these provide little or no ways for user intervention. As I will show, the possibility for user intervention enhances the usability of the model by allowing the user to fine-tune the model both for empirical

accuracy and interpretability.

A solution to this problem is proposed in the form of a new R (R Core Team, 2017) package, shinyproj, which allows its user to build models that achieve efficient trade-offs between model performance and interpretability. The package is novel as currently there do not seem to exist other tools that combine efficient model selection methods with an interface that simultaneously allows the user to explore the model space in order to more thoroughly understand the trade-offs in the model construction process.

The rest of this thesis is structured as follows. Section 2 introduces the topic of Bayesian predictive model selection. In this section, I review the the most common approaches for model selection, together with some practical variable selection heuristics. I also discuss some problems related to the approaches, such as the issues when there are strong relationships between the explanatory variables.

Section 3 discusses the issue of model interpretability. It reviews what makes models interpretable, why interpretability is important and then extends the idea of interpretability to the entire model selection process. I argue that a model and/or its parts can exhibit interpretability in degrees, with transparency as a main condition. While interpretability has yet to gain a conventionally accepted definition, in the context of model selection I take it generally to mean the accessibility of the information contained in the model. To be able to interpret the model means to be able to read not only the predictions it makes, but also the mechanisms behind the predictions. I argue that there are multiple important themes tied to interpretability, ranging from the ability to debug to public trust in algorithmic modelling. All of the issues provide reasons for interpretable models and model selection. Last, I discuss the ways interpretability could be extended into the whole modelling process and not just its result.

Section 4 combines the ideas from the two first sections and introduces a modern workflow for interpretable variable selection using the new tool shinyproj. As men-

tioned, shinyproj aims to fill a current gap in the model selection process, where there are significant, efficient tradeoffs to be made between interpretability and model accuracy by allowing the user to use their subjective information to edit the model. The functionality of the tool is discussed with an example application along with a discussion of its limitations. Finally, section 5 concludes the thesis.

2 Bayesian predictive model selection

Formally, model selection refers to the process of identifying a model m from the set of all available models M given a measurement (X, y). Model selection is usually concerned with comparing different candidate models rather than validating a given model as the former is often much simpler in practice.

Typically, model selection tries to achieve two related, but often separate goals. First, the model should provide some meaningful information about the process that is modelled. Second, the model should also be at least somewhat generalisable, ie. it should be able to make predictions about some previously unseen data.

A tractable model that explains the observed data perfectly might not have any predictive predictive power and correspondingly, there might be completely intractable models that predict perfectly, i.e. focusing solely on one of the goals does not guarantee any success on the other. On the other hand, there is often little sense in trying to interpret the learned model parameters or making inference based on the model that does not generalise at all. Therefore, model selection based on the predictive power of the model is a natural and widely used criterion and also the main consideration of this thesis. Following Piironen and Vehtari (2017a), this is later referred to as predictive variable selection.

The remainder of this section is organised as follows: First, section 2.1 reviews the most common methods for Bayesian predictive selection vaguely following Vehtari et al. (2012) and Betancourt (2015). Section 2.2 restricts the problem to variable subset selection and presents some algorithmic ways of finding suitable models among the set candidate models. Section 2.3 discusses the problems related to current approaches and motivates them with a few examples.

2.1 Model performance assessment

A commonly adopted approach for Bayesian model selection is to state it as a decision problem. The problem is to find a model $\hat{m}$ that maximises the utility, i.e.

$$\hat{m} = \arg\max_{m \in M} u(m, \tilde{y}), \tag{1}$$

where $u(\cdot)$ associates each model with a value, higher numbers implying higher utility and thus better models. As the interest is in models that predict well, it seems sensible that the utility is a function of the model and the future observation(s) $\tilde{y}$.

The desired criterion for a model is that it predicts well, and therefore the utility should somehow depend on the predictive density $p(\tilde{y}|m, X, y)$. Furthermore, it makes sense to demand that the utility function is proper, ie. that it is maximised (only) when the selected model equals the process that generated the data. As Bernardo (1979) suggests, the logarithmic predictive density

$$u(m, \tilde{y}) = \log p(\tilde{y}|m, X, y) \tag{2}$$

is often a sensible choice satisfying these criteria.

The future observations $\tilde{y}$ are unknown (by definition), and the equation 2 cannot be evaluated as such. A common remedy is to use expected value of the utility with respect to $\tilde{y}$ instead, yielding a criterion

$$\hat{m} = \arg\max_{m \in M} E_{\tilde{y}}[u(m, \tilde{y})], \tag{3}$$

where plugging in 2 yields:

$$E_{\tilde{y}}[u(m, \tilde{y})] = \int p(\tilde{y}) \log p(\tilde{y}|m, X, y) d\tilde{y}. \tag{4}$$

A problem that remains is how to evaluate the expectation over $\tilde{y}$? The most simple

way is to approximation the distribution of $\tilde{y}$ with a delta distribution at y, yielding

$$E_{\tilde{y}}[u(m, \tilde{y})] \approx E_{\delta}{}^{1} = \log p(y|m, X, y). \tag{5}$$

However, the problem with the quantity above is that as the model is fitted using the same data as what is used for the approximation for the future observations. This means that the estimate will, in general, be upward biased, thus giving overly optimistic picture of the model performance.

2.1.1 Hold-out Estimators

Typically one assumes that the density in (5) factorises, ie. the observations (X, y) are independent given the model m. This allows the data to be partitioned into two sets, namely the training and the test set. As only one of the sets is used for training the model and the other for evaluating the predictive density, the predictive performance is no longer overestimated due to using the same data twice. The approximation is therefore

$$\text{elpd}_{\text{holdout}} = \log p(y_{te}|m, X, y_{tr}) \tag{6}$$

where the subscripts tr and te refer to the training and test sets, respectively.

As one would expect, the bias-correction does not come for free. As noted, it is crucial that the observations at which the predictive density is evaluated (y_{te}) approximate the unseen observations ($\tilde{y}$) relatively well. Therefore, having a small test set means that the estimate of the predictive density is extremely noisy. On the other hand, if the size of the test set is increased at the expense of the training set size, one eventually runs into problems with fitting the model, again giving a suboptimal estimate of the performance.

[1] Following the convention in e.g. Vehtari et al. (2017), the approximations to the expected utility are referred to as elpd_, which could also be only proportional to the quantity as usually the scale itsef is not interesting.

One way to get more out of the n available samples is to use cross-validation. It partitions the data into K sets and fits the model K times, each time leaving out one of the sets and then using it to evaluate the model performance. The criterion becomes

$$\begin{aligned} \text{elpd}_{\text{cv}} &= \frac{1}{K}\sum_{i=1}^{K} \log p(y_i|m, X, y_{-i})^K \\ &= \sum_{i=1}^{K} \log p(y_i|m, X_{-i}, y_{-i}), \end{aligned} \tag{7}$$

where the subscript i refers to the i:th partition and the subscript $-i$ refers to all but the i:th partition.

The benefit is that now all the data points are used to evaluate the performance, thus improving the approximation. In addition, as (6) essentially averages over K training/test-splits, it will be more stable than (7) and having multiple terms to sum/average over enables one to estimate the variance of the estimate as well. Thus, stability-wise, the best choice is to use each observation as its own partition, often called leave-one-out cross-validation. In theory, this would lead to fitting the model n times, which might be expensive in some cases. Luckily, there are methods that make this computationally efficient such as the importance sampling approach presented in Gelfand et al. (1992).

2.1.2 Information Criteria

Another way of correcting for the bias in (5) (still assuming that the density in it factorises) is to add an explicit penalty term that penalizes models by their ability to overfit to the data. Traditionally, estimates of such form are known as information criteria.

Most straightforward and a widely used estimate is the Akaike information criterion (Akaike, 1974), which is formulated as follows

$$\text{elpd}_{\text{AIC}} = \log p(y|m_{mle}, X, y) - p, \tag{8}$$

where the subscript *mle* emphasises that the model is fit using maximum likelihood and p is the number of parameters. Thus, the more parameters in the model and moreover the mode complex the model, the more it is penalised. To be precise, the Akaike information criterion is the quantity above multiplied by -2 to on the same scale as the deviance of the model, another common summary of model fit, but it is omitted here for a clearer connection to the other estimates.

Designed originally as a tool for maximum likelihood estimation, there are reasons that make (8) somewhat suboptimal for Bayesian modelling. This is because the models are not estimated using maximum likelihood, but rather with (soft) constraints via the use of prior information. This makes p too harsh a penalty for models where prior information is used.

An alternative that is slightly more Bayesian is the Deviance information criterion (Spiegelhalter et al., 2002). It is similar to (8), but the maximum likelihood estimate is replaced with the posterior mean (although the authors state that some other summary, such as posterior mode or median could be also be justified) and rather than using the number of parameters as the penalty term directly, the penalty is estimated from the data. The Deviance information criterion is calculated as

$$\text{elpd}_{\text{DIC}} = \log p(y|m_{post}, X, y) - p_{\text{DIC}} \tag{9}$$

where the subscript *post* refers to the posterior mean. The penalty term is defined as

$$p_{\text{DIC}} = 2\left(\log p(y|m_{post}, X, y) - E_{m|X,y}[\log p(y|m, X, y)]\right), \tag{10}$$

where the expectation is taken with respect to the posterior distribution. The intuition with the penalty term is that if the model is severely overfitted to the data, the predictive density with the posterior mean will be higher than the expected density, penalising for overly complex models. In some cases, the posterior expectation is available analytically, but more generally it is often obtained by MCMC sampling or other approximate methods, or similar methods. In addition, the original authors

show that with approximately normal likelihood and negligible prior information (ie. flat priors), the penalty term reduces approximately to the number of parameters in the model, equalling to (8), as this also means that the posterior mean will equal to the maximum likelihood estimate.

However, even with (9) there is a concern that using just a point estimate (regardless of whether it is the mean, mode, or median), it will not fully capture the entire posterior. A fully Bayesian quantity that achieves this is the Widely applicable information criterion (Watanabe, 2010).

$$\text{elpd}_{\text{WAIC}} = \frac{1}{n}\sum_{i=1}^{n} \log E_{m|X,y}[p(y_i|m, X, y)] - p_{\text{WAIC}} \tag{11}$$

where again the expectations can be obtained by, e.g. MCMC methods. The quantity is similar to (9), but now the predictive density using the posterior mean is replaced with the expectation of the density over the posterior, and the density is evaluated pointwise. The penalty term is defined as:

$$p_{\text{WAIC}} = \frac{1}{n}\sum_{i=1}^{n} \left(\log E_{m|X,y} p(y_i|m, X, y) - E_{m|X,y}[\log p(y_i|m, X, y)]\right), \tag{12}$$

which in turn resembles 10, but is averaged over the posterior to capture the full posterior uncertainty. In addition, Watanabe (2010) also proves that (11) is also asymptotically equal to (7), giving a justification that it indeed approximates the expected utility as defined in (3).

There are also other quantities, such as BIC, that are relatively widely used and, somewhat confusingly also called information criteria but which have goals other than maximising the predictive accuracy. Since the main focus of this thesis is on predictive model selection, the details are omitted here.

2.1.3 Reference model based approaches

Section 2.1.1 considered estimators that hold out part of the data from model fitting and then use the held out part to approximate future observations. Instead of trying to find a model that best fits the future observations (on expectation), the reference model based methods first form what is called a reference model, and then use that as a reference point for the model comparison. The idea in reference predictive model selection, as presented in Vehtari et al. (2012), is that the reference model represents the best possible model using all the information and data that is available. As the reference model is, by definition, the best possible model it can be used as a reference for model comparison, for example in order to assess whether some simpler model performs sufficiently good.

In terms of the objective function for the model comparison, this simply means substituting the distribution of future observations $p(\tilde{y})$ with the predictions of the reference model $p(\tilde{y}|\hat{m}, X, y)$ (compare with (4) and (5))

$$\text{elpd}_{\text{ref}} = \int p(\tilde{y}|\hat{m}, X, y) \log p(\tilde{y}|m, X, y) d\tilde{y} \tag{13}$$

The quantity (13) is often formulated in terms of KL-divergence (Kullback and Leibler, 1951). This is because

$$\begin{aligned} &\int p(\tilde{y}|\hat{m}, X, y) \log p(\tilde{y}|m, X, y) d\tilde{y} - \int p(\tilde{y}|\hat{m}, X, y) \log p(\tilde{y}|\hat{m}, X, y) d\tilde{y} \\ &\qquad = -\int p(\tilde{y}|\hat{m}, X, y) \log \frac{p(\tilde{y}|\hat{m}, X, y)}{p(\tilde{y}|m, X, y)} d\tilde{y} \\ &\qquad = -\text{KL}(p(\tilde{y}|\hat{m}, X, y) \| p(\tilde{y}|m, X, y)), \end{aligned} \tag{14}$$

where $\text{KL}(p\|q)$ refers to the KL divergence between the distributions p and q. Note that the quantity that is subtracted, is constant with respect to any model m that is being compared and therefore it does not affect the model comparison. Therefore, finding the best model with respect to criterion (13) is equal to minimising the KL divergence from the reference model.

A formulation of this problem that is restricted to variable subset selection is the projection approach (Goutis and Robert, 1998). In the projection approach the idea is that the reference model is the model using all the available variables and the sub-models that are being compared use a subset of the available variables.

Intuitively, minimising (14) for a fixed subset of variables linearly projects the information from the full model to a sub-model. In other words, it results to the coefficients for the variables in the sub-model that yield predictions as close to the reference model as possible.

While the KL-divergence from the reference model to a sub-model is not analytically available, it can be approximated with MCMC as proposed in Dupuis and Robert (2003). In addition, the minimisation of (14) between the reference model and a sub-model is a convex problem for exponential family models and therefore easily solved via readily available optimisation methods, which is why it is originally proposed as a model selection tool specifically for generalised linear models.

2.1.4 Other approaches

Bayesian theory offers a natural construction that allows one to avoid the model selection completely. In Bayesian model averaging, one obtains the posterior probabilities of each model being true via the Bayes rule

$$p(m|y, X) = \frac{p(y|m, X)p(m)}{p(y|X)}, \tag{15}$$

where $p(m)$ refers to the prior probability of model m being the true model. The predictions for future observations are obtained by averaging over all the candidate models

$$\begin{aligned} p(\tilde{y}|y, X) &= \sum_{m \in M} p(\tilde{y}, m|y, X) \\ &= \sum_{m \in M} p(\tilde{y}|m, y, X)p(m|y, X). \end{aligned} \tag{16}$$

The benefit with the formulation above is that in theory it should never do worse than discrete model selection, and it indeed seems to work well (e.g. Raftery and Zheng (2003)), especially if the true data generating process is within the set of candidate models. However, when the true model is not included in the set, the method is shown to not work optimally (Yao et al., 2018). Furthermode, calculating (16) can be slightly awkward in practice.

In order for the formula to be exactly correct, the set of all models M should contain every possible model that has a nonzero (posterior) probability of being true. Therefore, as one can consider and evaluate only so many models, using the equation (16) can be somewhat hard to justify exactly.

In addition, the posterior probabilities of models being true can be sensitive to assumptions that are not testable with the available data. For example, assume that there is a parameter of a model, the true magnitude of which is around 5. The true magnitude of the parameter is not known for the modeller and it is given a flat prior. Now, whether the prior has a scale of 10^5 or 10^8 should not have any practical effect on the actual performance of the model, but it will essentially be a multiplier to the posterior of the model and thus affect the contribution of the model to the Bayesian model average significantly.

In practice, there also exists a multitude of ad hoc measures of model accuracy that, while probably originally motivated by rigorous reasoning, are often used mainly because they are so commonly used and/or because of their mathematical convenience.

A prime example of this is the mean squared error (multiplied by -1 to formulate it as an utility function)

$$u(m, \tilde{y}) = -(y_m - \tilde{y})^2, \tag{17}$$

where y_m denotes the predictions of the model m. A clear benefit of the mean squared error is its mathematical convenience – it is extremely fast to evaluate, convex and differentiable everywhere. Possibly because its simplicity, it is probably one of the

most widely used utility/loss functions in supervised learning and statistics in the case of a continuous target.

Its connection to the log predictive density discussed in the beginning of this section is also easy to see. It is proportional to the log predictive density of a Gaussian random variable (with known variance), which means that for Gaussian random variables maximising (17) is equivalent to maximising (2). Therefore it can also be viewed as a kind of Gaussian approximation. However, also because of this all the problems related to using the same data for fitting the model and evaluated it with the same data apply, and corrections such as hold-out estimators or information criteria are needed.

In ideal cases, there is some domain knowledge that allows the construction of a function that best fits the problem at hand. For example, one might know that for the problem at hand, it is much more dangerous to err on the positive side than on the negative side, and construct an appropriate utility function for this. Therefore, it is clear that the logarithmic predictive density is not always the best possible form of the utility function. However, in the general case where there is no reason to think otherwise, it is probably a good first choice.

2.2 Variable selection heuristics

One of the most concrete and simple examples of model selection is the variable subset selection. In variable subset selection, the task is to try to find a subset of the variables that are most suitable, for example a subset of the variables that is as small as possible but still does not lose much of the predictive power compared to a reference model.

Traditionally one major reason for variable selection has been that many statistical methods have trouble distinguishing the relevant effects in the data and furthermore overfit to the training data when the number of available variables is large compared

to the number of observations. With modern Bayesian methods (as well as some frequentist methods, such as e.g. the Lasso (Tibshirani, 1996)), this problem can be often at least greatly mitigated, if not completely solved via regularizing priors (e.g. Gelman et al. (2008)).

Regularizing priors do not, however, eliminate all the problems. One of the other reasons for variable selection is that in several cases there are some costs associated with the data, e.g. computational costs related to fitting the model or measurement costs related to collecting the data. In addition, models with fewer variables are easier to interpret, which makes small number of variables preferable.

As variable selection is just a subset of model selection, all the performance evaluation methods presented in section 2.1 are directly applicable. One of the problems is though that as the number of candidate variables increase, the number of variable combinations grows exponentially. This makes exhaustive search search computationally infeasible even with a moderate number of candidate variables and therefore more efficient strategies are needed.

2.2.1 Stepwise regression

While the idea of stepwise regression is in theory slightly more general, in practice it refers to either forward selection or backward elimination. In forward selection, one starts with an empty model and adds variables to the empty model one at a time, each time adding the candidate variable that maximises the given objective function, such as KL-divergence to the reference model. In backward elimination, one starts with a model with all the candidate variables and removes variables one at a time, each time removing the variable that decreases the given objective function the least. In general the results of the algorithms are relatively similar, but for example in sparse settings, ie. settings where one assumes that only a small fraction of the variables are relevant in terms of the model performance measure, the results might differ.

One of the clearest benefits of stepwise regression methods is their relatively modest computational complexity. As each time they consider adding or removing one of the variables and terminating as all the variables have been added or removed to the model, the running time grows only quadratically with respect to the number of variables.

However, a concern related to stepwise regression is that they are relatively susceptible to overfitting. As they essentially select the best model among a quadratic number of comparisons, it is highly likely that the model selected will tend to be overestimated. Therefore, checking which of the selected variables have statistically significant coefficients, for example, would not be valid, as they are selected to the model because they have an effect on the output.

With predictive model selection, this is not such a big concern. This is because the target of the variable selection is not to for example perform hypothesis testing on the model but rather simply find a model that predicts well. Still, the predictive performance of the model will be overestimated because of the aforementioned reasons. Strategies to mitigate this will be discussed in section 2.3. In addition, the reference model based approaches will also somewhat attenuate this, because we are not fitting the model on the data, but rather fitting the model to the reference model, which in turn is by assumption not overfitted to the data.

The stepwise regression methods do also seem to work well in practice in combination with the projection methods (Piironen and Vehtari, 2017a).

2.2.2 L_1 Search

Another popular method to perform variable selection for regression models is achieved via the lasso estimator (Tibshirani, 1996). The idea of the lasso is to add an L_1-penalty to the log-predictive density, which would mean that predictive utility would be of the form

$$\log p(y|m, X, y) - \lambda \|\beta_m\|_{L_1}, \tag{18}$$

where β_m refers to the regression coefficients of model m, $\|\cdot\|_{L_1}$ to the L_1 norm, and λ a multiplier for the penalty (the higher the λ, the more the variables are shrunk towards zero).

The intuition is that the L_1-penalty shrinks all the variables regardless of their magnitude, shrinking the unimportant variables to zero, keeping the important variables unpenalised. In addition, the solutions of the lasso will be equivalent to using a laplace prior on the regression coefficient in Bayesian linear regression with λ as the precision parameter, giving further aid in the interpretation.

As the magnitude of λ defines the amount of 'total shrinkage', by varying it one can essentially an ordering for the variables, $\lambda = \infty$ setting all the variables to zero, $\lambda = 0$ setting all the variables to their unpenalised values and λ values in between setting some of the variables to zero. Therefore, by evaluating maximising (18) with a monotonous sequence of λ's one obtains an ordering for the variables, giving lasso its role as a variable selection heuristic.

2.3 Problems

2.3.1 Overfitting and selection bias

The predictive performance of the model tends to be overestimated, because the estimate is actually calculated for the best variable combination given the data, rather than any given variable combination. Another way of putting this is that after we have used the data to find the best candidate model for a given model size, we can't use the same data to validate its performance. Section 2.1 presented some ways to assess model performance. While some of the methods claim to overcome the problem of overfitting, the problem is that they are only able to assess the performance of a single model. However, there is another level of overfitting related to the selection process. As we use the same data to select the best model and evaluate the performance of the models, the performance of the best model is likely

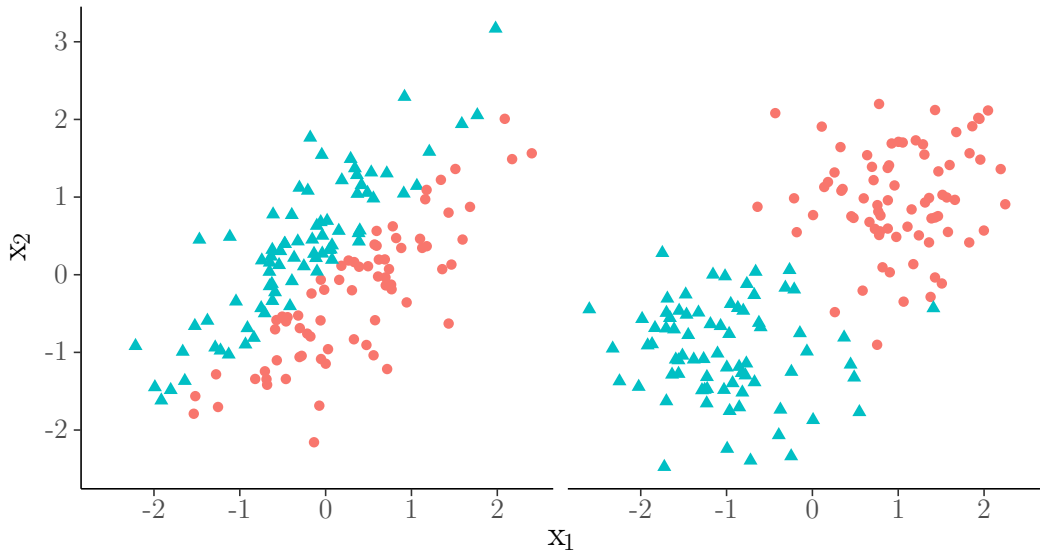

Figure 1: Two examples of tricky cases for variable selection in binary classification.

to be overestimated.

One solution that can and is used for this is to use another form of correction, e.g. perform a cross-validation for the entire search process (Vehtari et al., 2012).

2.3.2 Correlated variables

A simple scenario in which most general purpose variable selection methods – including the aforementioned two – have problems are related to the correlation structure between the candidate variables. Figure 1 presents two such examples.

In the left figure, the variables explain the data perfectly when both of them are included in the model but adding any one of them does not relate to the output at all. In that case, the selection method might add the given variables to the model well after other, less important variables, if the other variables – when considered in isolation – have even a negligible effect. In the right figure, adding any one of the variables is enough after which the second variable does not provide information. The problem is that the choice is extremely sensitive to the observations near the decision boundary.

Both problems demonstrate that some external information is needed for robust and accurate inference. Assuming that the correlation structure is known beforehand, a straightforward generalisation to solve the former problem, namely the Group lasso (Yuan and Lin, 2006), in which the variables are simply added in groups instead of one by one. Similar group-wise comparison also trivially extend the stepwise regression methods.

2.3.3 Stopping criteria

The discussed selection criteria in combination with the search heuristics, such as the projection method with the forward selection or L_1-search, do provide a well defined solution for any given model size (or regularisation value). What they do not explicitly provide, however, is information of which sub-model is 'close enough' to the full model, or conversely, when adding additional variables to the sub-model do not really provide enough predictive power to justify their adding to the model.

For the L_1-type methods, the model size is typically selected by cross-validation on which sub-model size leads to best performance on the test set. This, however, relies on the assumption that one wants to find the best model and not some small, but still a good one. If the full model is not overfit to the data at all, as is assumed e.g. in the projection methods, this yields to always selecting the full model.

For the projection method, Dupuis and Robert (2003) propose a choosing a bound of maximum acceptable loss in explanatory power $e(m)$. This is achieved by scaling the original optimisation criterion of KL-divergence from the sub-model to the full model (14) by

$$e(m) = \frac{\mathrm{KL}(m\|\hat{m})}{\mathrm{KL}(m_0\|\hat{m})}, \tag{19}$$

where m is the model being compared, m_0 is a null model with only an intercept term and $\hat{m}$ is the full model.

As the KL-divergence is monotonous in the number of variables, the above can

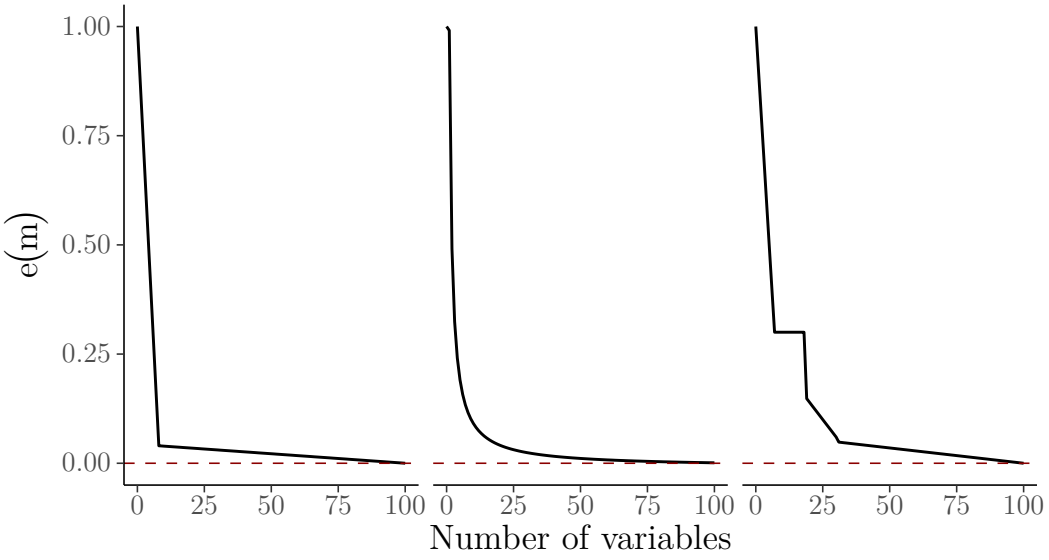

Figure 2: Possible realisations of the loss in predictive power as a function of the model size.

be interpreted as a fraction of information projected from the full model to the sub-model and e.g. Vehtari and Lampinen (2004) use this heuristic to stop the forward search when the above criterion is at most 0.01. The formal justification and further intuition for the criterion can be found from the original article (Dupuis and Robert, 2003).

This procedure is good in that as the cutoff is selected independently of the selection process, it does not cause any overfitting. While having the cutoff around 0.01 is good for stopping the search early as no notable gains will be achieved by adding additional variables, the model with 0.01 loss in explanatory power is rarely the best one in terms of the tradeoff between reducing the number of variables and losing in explanatory power.

In practice, one would often like to use different cutoff points dependent on the shape of $e(m)$ as a function of the model size. Figure 2 illustrates a few scenarios that demonstrate the difficulty in choosing the cutoff independently of the data.

In the leftmost plot, it is likely that a model close to the kink would be preferred, as before the kink there is a clear benefit in adding variables and after the kink, adding

additional variables does not increase the performance of the model. Furthermore, it would also not be too hard to find this kink automatically, and in some cases the optimal model size is quite clear regardless of what the actual task is.

However, in many occasions the size selection is deeply related to the actual data at hand. The middle plot shows a situation where halving the loss at any point requires doubling the number of variables in the model. The model size selection is now effectively an explicit tradeoff between the loss in performance, and therefore will depend on other factors than are found from the data, e.g. measurement costs or costs of making a wrong decision.

The rightmost plot further emphasises how the task of automatically selecting model size in the a general setting where the shape is hard. The shape of the curve likely rules out some model sizes, such as the region where the the curve is flat. Still, to really get the best model in the particular situation, some human intervention is needed.

3 Interpretable model selection

As discussed in the previous section, the problem of model selection is hard to automate, and in general there is no one metric that is suitable for all model selection problems. In practice, the best results are often gained by adjusting the model or model selection criteria to the problem at hand, which essentially means using outside information.

An immediate solution is to extract all the information from the user beforehand, combine it with the data and then perform the model selection given the data and the outside information. However, in practice, this naive approach runs into at least two problems. First of all, it is difficult to know what to ask the user for the outside information to be considered sufficient, not to mention complete. Second, the modelling process itself often yields new insights about the data or the target system that is being modelled. In addition, it might be extremely difficult to formalise the user information for the modeller to be able to combine it with the data (for work related to this, see e.g. the SHELF software package (Oakley and O'Hagan, 2016; Zapata-Vázquez et al., 2014)).

On the other hand, given that the models are interpretable throughout the model building process, the user can provide the information on demand. However, the interpretability requirement, comes with a cost, as obviously not all models are easily interpretable.

In addition, the issue of model interpretability is important because it benefits the user of the model by making the model easier to use. Especially if the model is used to describe the interaction of individuals, interpretability is also in their interest, because the use the model is put to often affects their decisions or available courses of action. Indeed, machine learning models are used more and more in public decision making (for examples, see e.g. Zeng et al. (2017) for recidivism prediction or Letham et al. (2015) for stroke prediction). Given that such decisions have an increasing effect on individual lives, the EU has also noted the need to set regulations concerning the

use of models in public decision making.

While the majority of the EU General Data Protection Regulation (GDPR) is about privacy and the collection of personal information, it also contains the section 'Article 22: Automated individual decision-making, including profiling'. Enforced from mid-2018 onwards, Article 22 contains concepts such as 'a right to a meaningful explanation' and 'a right to intervention', making clear that the models used in public decision making or used by institutions such as banks need to be interpretable and explainable. The users (of the model) will need to be able to explain how the model works and why it works the way it does. Indeed, in some situations, using complete black-box models might be considered illegal.

The rest of section is organized as follows. Section 3.1 continues to consider factors that make models interpretable. Section 3.2 attempt to further argue why interpretability important and section 3.3 extends these ideas to the entire variable selection process.

3.1 What makes a model interpretable?

3.1.1 Interpretability as transparency

Interpretability is a characteristic of any given model. A model or its parts can be interpretable to a certain degree. In general, interpretability enables the modeller or the user of the model to see how and why the model works the way it does. If a model is interpretable, the effects from the interaction of its parts are clear and the model as a whole is tractable.

Several authors relate interpretability with transparency (e.g. Lipton (2016); Lou et al. (2013)). In the definition presented above, transparency is a necessary condition for interpretability, ie. non-transparent or opaque models are not and cannot be interpretable.

Ideally, the models will also allow for counterfactual what-if reasoning. In other

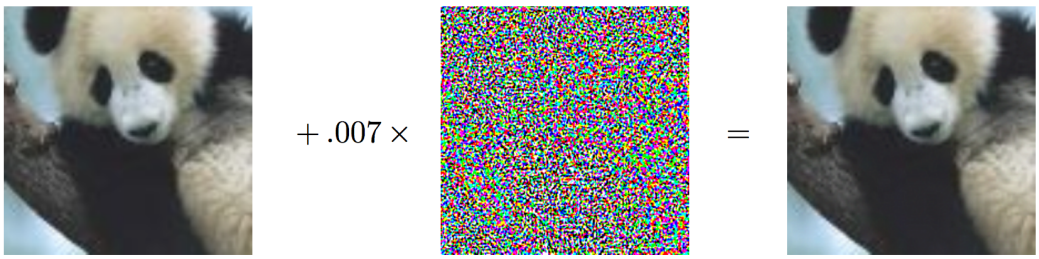

Figure 3: Example from Goodfellow et al. (2015). The leftmost image is classified as a panda (with 57.7% confidence). However, when the middle is added to it, the result is classified as a gibbon with 99.3% confidence.

words, given the fitted model, the user should be able to answer questions such as 'How much would the values of the variables need to change that the prediction would change this much?' or conversely 'How much will the prediction change if this variable changes?'. Furthermore, the model can be used in the construction of explanations of real-world phenomena via the what-if questions it allows. However, the possibility for formulating what-if questions on the basis of the model is not a condition of interpretability, but it can enhance the interpretability of the model. Mostly what-if reasoning done on the basis of the model benefits from the model already being interpretable.

As an example, consider deep neural networks that are often (e.g. Lou et al. (2012)) argued to be very difficult to comprehend. In fact, not only are e.g. sufficiently complex neural networks hard to interpret, but in some situations it does not even make sense to try to interpret them. Figure 3 illustrates one situation, where the non-transparency causes real problems.

The network classifies the leftmost image as a panda, and the rightmost as a gibbon, even though there are no perceivable differences between them. By making slight adjustments, the neural network changes from classifying both images as pandas to making a pretty big mistake, at least where animal classification is concerned. While these kinds of examples can seem quite ad-hoc and irrelevant, they illustrate the issues with opaque models well. The examples do need to be very carefully crafted, but they show how particular models make mistakes with no visible source for the

mistake made. In addition, these kinds of issues generalise relatively well: If one were to retrain a different, but relatively similar model on the same data, it would likely make the same mistake (Goodfellow et al., 2015). While problems with animal picture classification might not seem to pose a threat to the society, it is not hard to think of a similar example involving e.g. self-driving cars.

On the other hand, for example the US credit-scoring company FICO apparently trains their model specifically with interpretability in mind (Lipton, 2016). Could the same sort of 'gaming' as with the animal classification example work here? Again, slight external adjustments could be made in order to fool the mode. In fact, FICO openly acknowledges that there is a possibility that the rating could be gamed and even provide advice on how to improve one's credit score as given by the algorithm. However, the fact that the information is public makes it less likely that any serious large scale exploitation could be underway, as FICO will probably be on the lookout for gaming attempts. In general, given that the details of the model are transparent, problems like this will eventually get detected.

3.1.2 Linearity and Sparsity

There are two concepts that several authors (e.g. Kim (2015); Ribeiro et al. (2016a); Zeng et al. (2017)) hold as key for interpretability, namely linearity and sparsity.

Linearity is typically a good guarantee for a possibility of what-if reasoning. This is because as the output is always a linear function (possibly with a link function) of the inputs, a change in output can be obtained by simply multiplying the changes in the inputs by their respective weights. Even though the possibility for what-if questions is not a condition of interpretability, it can enhance the interpretability and above all, the usability of the model.

However, linear models are not always more interpretable (as argued in e.g. Lipton (2016)). A possible concern is the existence of strong correlations between the variables. In that case, changing a the value of a variable or a small set of variables

independently of the others might not be realistic. In this case the interpretability can be retained retained with sparsity. If there are only few (nonzero) inputs that effect the dependent variable, the change in output is easy to follow if one or some of them are changed.

Neither linearity nor sparsity alone is enough to guarantee interpretability. A conceptual problem related to more non-linear models such as Gaussian processes or multi-layered neural networks is that the relationships between the predictors and the outcomes are inherently local, meaning that the relationship varies depending on the input (which of course is also sometimes a desirable property if the phenomenon is of that nature). This makes models with only a few variables difficult to summarise, especially in contrast to linear models where the global relationship can be summarised a lot more easily.

This is of course not to say that non-linear models are impossible to interpret. Indeed, a lot of research and successful attempts have centred around making these more complex and rich models more interpretable (e.g. Duvenaud et al. (2011); Ribeiro et al. (2016b)). However, here the focus is kept on models that are designed to be interpretable, rather than on models whose interpretability is improved in order to achieve certain goals.

3.2 Why is interpretability important?

In general, interpretability is important for technical reasons and reasons related to the use of the model by public of private institutions. The purpose of a model is to model a target system, usually a relationship between given variables. Interpretability is a characteristic of the whole model as well as its parts, that makes possible for the user of a model to explain how and why the model or its parts work the way they do. If the variables being modelled concern the lives of individuals or social wholes, interpretability is a condition for accountability and fairness in the use of the model by public or private institutions, such as banks or companies.

3.2.1 Model Debugging and Accountability

From the technical perspective, an immediate benefit of interpretable models is an increased ability to debug the models. When a problem such as a faulty prediction occurs, interpretability makes preventive action possible. When the operating mechanisms of the model are visible and explainable, it is also easy to see where and what interferes with how the model is supposed to work. If the model is a black-box in the sense that it does not distinguish between the different causes for a certain prediction, it might be difficult to find the source of the problem. While there is also extensive literature on explaining the predictions of any model (e.g. Ribeiro et al. (2016b)), for interpretable models the process is much more straightforward.

In interpretable models, outside information is often key. Theory is often available, such as in the context of medical diagnosis. Based on that information it is often possible to observe non-sensical predictions by the simple observation of for example regression weights.

Related to knowing the mechanisms within the model is the 'right to intervention' provided by the EU GDPR. If the model is a black box and the source of the problem, be it from a bug or from wrongly estimating the value of some variable, is unknown, interventions are time-consuming and costly. Interpretability is key in the user being able to formulate hypotheses about why the model does or does not work. Interventions testing these hypotheses can then be used to formulate explanations about the model and its mechanisms. When designing and implementing an intervention, the party responsible must understand the reasons behind the decisions of the model in order to succeed. The need for understanding the model is amplified when models are used more in public decision-making.

When using a model, human intervention is also key because of the nature of models. Models systematise errors in a different way than humans: they do not forget the instructions given to them, but they are also unable to divert their course of action from them. If errors are encoded into the model in the instructions, the model will

make errors systematically and it will not be able to correct itself unless intervened upon. A transparent model often makes these kinds of mistakes quite visible, which in turn makes human intervention much easier.

As Gandy (2010) notes, 'most of the time, persons who have been victimised by a routine system error will not know precisely if, when, or how they have been discriminated against'. This is an issue related to the nature of errors made by models – they are often subtle and nearly invisible, especially if the user of the model is not aware that institutions or the tools they use can be discriminatory towards certain subpopulations.

Before the issue of trust in algorithms and algorithmic models (further discussed in section 3.2.3), there is the issue of accountability, also related to modelling failure, debugging and interventions. If there is no way of knowing what caused the model to fail, it is also harder to find out the person responsible. While this is an open (legal) problem, interpretability provides at least one simple solution to this. As long as the model is interpretable to someone, a person can be assigned to be 'responsible of the model behaviour'. As an example of this, Tesla's 'Driver Assistance features' require the driver to always keep their hands on the wheel, as the driver is really accountable even if the model does the actual driving.

In short, interpretability is key in a number of issues related to modelling failure. First, there is the issue of debugging and finding out what caused the model to fail. Interpretability makes diagnosing the failures of the model much easier, as well as interventions on the model. Last, interpretability ensures accountability, at least to the extent where it is possible for modellers and the users of the model to answer ethical questions with practical solutions.

3.2.2 Fairness

Fairness is an issue related to the use of algorithmic models in public decision making. While accountability is the question of who is responsible for the mistakes the model

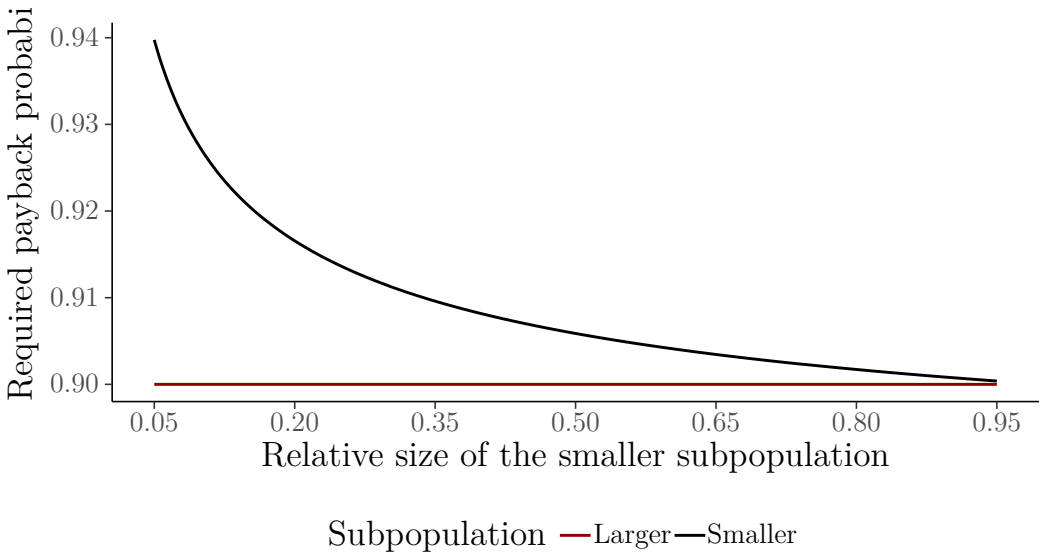

Figure 4: Illustration of the of the uncertainty bias. The plot shows the required payback probability for each subpopulation to have same lower 0.05-quantile as the larger group when the size of the larger subpopulation is 1000.

makes, fairness is a characteristic of the model itself. Even in model building, fairness is not an easy concept to define. However, much of what makes a model fair can be characterised by the consequences its use has. Interpretability in a model may not prevent unfair models from being constructed, but it again makes intervention upon the unfair models easier, more time efficient and less costly.

One consequence of unfair models can be discriminatory decision making, where the source of discrimination is a biased model. This is a a well studied (e.g. Calders and Verwer (2010); Zhang and Neill (2017)) issue that GDPR also deals with. Models used in decision-making need to be non-discriminatory, which is very difficult, if not impossible, to ensure in practice. For example, in order to combat discrimination based on religion, one might decide to omit information about religion when building the model. However, if there are any variables that are correlated with ethnicity in the data (such as zip-codes, if places of worship for different religions are located in different areas but close to their ethnic or religious communities), the outcomes of the model will effectively be correlated with religion.

Another well known example related to fairness is what Goodman and Flaxman (2016)

call uncertainty bias, observable especially in credit scoring. Again, interpretability proves to be a useful characteristic for a model to have.

A lender might base the decision to give a loan on the lower quantile of the prediction that the debtor actually pays back. However, there is naturally less data from some subpopulations than others. This means that for subpopulations for whom there is less data, the mean payback-rate has to be higher for them to be eligible for a loan. Figure 4 illustrates the situation. The figure shows what the true probability of loan payback needs to be for each group to achieve the same 0.05-quantile. For example, if the size of the smaller subpopulation is 10% of the larger, they would need their actual payback rate to be 92.7%.

In practice it is very difficult to ensure that the models are non-discriminatory and even predictive models intended to be fair might turn out discriminatory (Calders and Žliobaitė, 2013). As (Žliobaitė, 2017) concludes, this is mainly due to biased data. If the data used to train the model is already biased, it is likely that the trained model will reflect these biases as well.

However, when the model is easily interpretable, it is possible to examine the model and address the possible discriminatory biases as they appear. This is true in cases where the data is biased in quality but also in cases like credit scoring, where the data is not inherently biased in quality, but in quantity.

3.2.3 Trust

In addition to debugging, accountability and fairness, trust in models is an issue related to the interpretability of models. When the use of models to aid in decisions in every day life increases, it becomes more and more important that the models that are in use can be trusted. Indeed, several authors list trust as one of the most important reasons (e.g. Kim (2015); Ribeiro et al. (2016a)) for interpretable models.

On one hand, there is the problem of trusting models that should not be trusted,

such as discriminatory ones. These also include models with technical faults, such as in the case revealed by Eklund et al. (2016). The authors find that the software used in essentially all fMRI-studies has contained a bug for 15 years, causing inflated false-positive rates. As the authors state, the methods had never been validated using real-world data, causing the problem to go undetected.

On the other hand, there is the issue of not trusting models enough. Most importantly, the party responsible of the implementation must be able to trust the model. A more serious example illustrates the problem: the automated safety system of the Three-Mile Island nuclear power plant had warned a human operator to shut down the system but the operator did not act on the information because they did not trust the automated safety system to estimate and predict the risk of accidents correctly. (Freitas, 2014).

Thus, it is important that people whose decisions the models influence to trust the model. For example, Veale (2017) documents cases where analysts at a tax agency are much more likely to adhere to recommendations made by an internal tool when they understand what it does and how it works.

The problem of public trust in algorithmic models is illustrated by a case in the US, where there has recently been discussion over whether a crime recidivism algorithm developed by a private company, Northpointe Inc., is fair. The case is also an example of how fairness and trust are inherently linked in models and their use in decision making by public institutions or private companies. The original article (Angwin et al., 2016) accuses the company of being biased against black defendants. The company responded (Dieterich et al., 2016) by stating that according to the metrics used by their criteria, the predictions are fair, but that the metrics pointed out in the accusations were selectively chosen to make the predictions look unfair. The issue has raised a lot of academic interest (e.g. Chouldechova (2016); Zhang and Neill (2017)) and the conclusion seems to be that whether the algorithm is biased is a matter of chosen metric. However, much of the concern is also related to the matter that the algorithm is proprietary and therefore not available to public scrutiny. As

in other cases, the issue of faulty models extend to fairness and trust, and ultimately accountability.

3.2.4 Information

A final, possibly most obvious reason is that interpretability is important for its own sake. Most of the time, models are not built just to build models, or just to black-box-predict certain outcome, but rather to extend the understanding of the phenomenon or target system modelled. In some cases, models can function as or help formulate explanations about real-world events and mechanisms. In image classification scenarios, it might not be particularly important to understand why one of the images is classified as a panda and the other as a gibbon. In most cases, however, such as when aiding medical diagnosis or predicting crime recidivism rates, it is also often equally if not even more important to also understand the actual process rather than to just obtain black-box predictions.

As an example of why understanding the actual phenomenon might be significantly more important than obtaining predictions, consider a model trained to predict probability of death from pneumonia as discussed in Caruana et al. (1999). The model learns from the data that asthma is associated with a lower risk of death from pneumonia. In reality, however, the risk is low because in the data, the patients with asthma have received a lot more treatment than others. Of course, if the model were deployed (ie. asthma patients would be sent home earlier than others) without human intervention, the resulting outcomes would eventually provide enough evidence that the model is faulty. Often, however, the cost of fixing problems in the models this way is simply too high.

3.3 Interpretable model selection for interpretable models

What does interpretability mean in the context of model selection? Interpretability in model selection means the interpretability of the steps taken to build and select the model. Understanding the heuristics used and decisions made throughout the modelling process will increase the usability of the model, because understanding the model again increases the user's capacity to plan and implement interventions on the model if it does not function properly or if it turns out that there is in fact a better model available for the task at hand.

An open question with the interpretability of model selection is automating the modelling process. There are some attempts to automate the entire modelling process (see e.g. the Automatic Statistician (Lloyd et al., 2014)), but in general, as discussed in section 2, the problem of model selection is not solved. There are several methods that try to extract the information from users as efficiently as possible, such as interactive clustering. However, as argued in Kim (2015), such methods are typically not designed to be interpretable, meaning that the while the user has helped build the model, the result might still not make any sense to the user.

For reasons discussed in 2.3, a thorough model selection task will often require some sort of intervention from the user. As Ribeiro et al. (2016a) notes, the issue is not of finding the best interpretable model, but sometimes different models provide different information which might also work as a decision criterion.

For the user to make informed choices, it is necessary that they understand the models that are being compared, ie. the models need to be interpretable. In short, this means that the user needs to be presented models of the type described in section 3.1 along with model performance metrics such as described in 2.1.

Another requirement for interpretable model selection is that the scope of the task given to the user is limited enough. In model selection the user can quickly become overwhelmed by the number of possible choices in the selection process. To illustrate

what this means, consider a variable selection task with k candidate variables. There are now 2^k possible models that should be compared. Now, even with $k \geq 8$ or so, it is quite clear that no matter how simple the models, it is not really possible to parse through them.

An immediate solution for this is to use some algorithmic technique to first prune models that are not likely to be useful, for example using forward selection, after which there are only k models left. In addition, by plotting the performance of the models such as in figure 2, it is likely that the user can quickly narrow the set of interesting models to a small subset of them, based on the shape of the curve.

A third requirement is that the model selection heuristic itself needs to be transparent enough. For example, a user aware of the path-dependency of the forward selection might be able to diagnose and fix problems such as the on presented in figure 1.

Furthermore, as discussed before, in several cases model selection is done not only to build a model that predicts as well as possible, but also one that enables the user to understand the actual phenomenon. In this case, while the algorithmic model selection process is not able to distinguish between variables that have equal predictive power, there might be a vast difference in the importance of one over the other for the user.

Another thing to note is that typically variable selection methods would prefer smaller models to larger models given that they perform equally well. For interpretability, this is not always the case. Elomaa (1994) documents how medical experts find more complex models in some situations more interpretable. In the paper he describes a case with *Nephropathia epidemica* patient records and a model with a single predictor, fever. Even though this overly simplistic model performs quite well on the data, the medical experts do not really find it helpful at all, as they do not understand it, and prefer a more complex model.

4 shinyproj

Section 2 reviewed model selection criteria and section 3 considered the issue of interpretable model selection. This section combines the findings of the previous section into a new tool to perform interpretable variable selection.

Traditionally variable selection is regarded as a linear process: obtain the data, specify the model space, run the variable selection algorithm to find the best variable subset and finally obtain a model. This section describes a modern workflow for interpretable Bayesian variable selection for generalised linear models via a new R package shinyproj[2].

The idea of shinyproj is to transform the variable selection process into a more iterative one by combining the model diagnostics with a user interface that enables exploration of different models. The first and possibly the most important task is to communicate the performance of the model in comparison to other possible models. The user interacts with the model to increase it's interpretability while the predictive performance comes from the algorithm.

In practice, the workflow is as follows: After fitting the full model, the user explores the model space with a guidance from the selection algorithm; given a suggested model, the user examines whether the model is sensible and performs well enough in relation to the other possible models. If not, the process is continued until a sufficient model is found.

For variable selection, the package uses the projection predictive method (as discussed in section 2.1) which is implemented in the projpred[3] R package. The choice of the method is based on Piironen and Vehtari (2017a), in which the method is found both to perform really well on real and simulated data as well as be less vulnerable to overfitting than other available methods. To be more precise, the projection approach is found the most efficient (with a low variance) way of achieving good

[2] Available at `https://github.com/paasim/shinyproj`.
[3] Available at `https://github.com/stan-dev/projpred`.

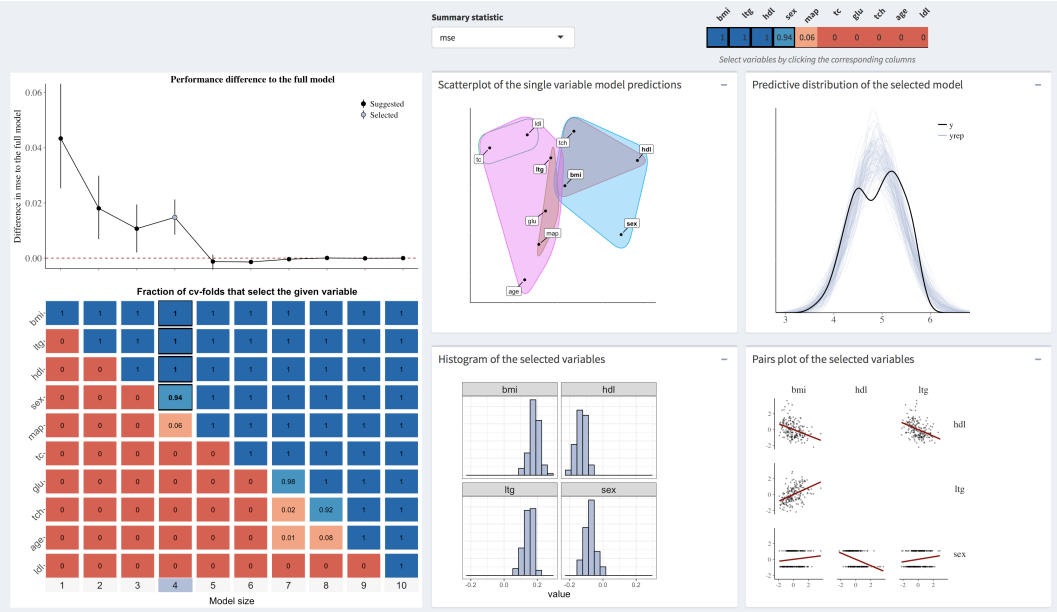

Figure 5: The shinyproj user interface.

accuracy-model size tradeoffs among the most common Bayesian variable selection methods.

Furthermore, shinyproj makes the interpretable variable selection tractable for the user by dividing the task into two manageable parts. The shinyproj user interface is shown in figure 5. First, the user is presented with information about the tradeoff between the model performance and the model size along with a metric related to the uncertainty of the variable selection process (the plots on the left-hand side). After a suitable tradeoff for the model size and performance is found, the user is allowed to diagnose and make more local edits to the selected model (using the information from the plots on the right-hand side). The local edits allow user to further tune the model according the particular needs to increase interpretability. As the user adds or removes variables to or from the model, all the plots in figure 5 update accordingly. In this manner, the task of model selection is essentially split into a choice between at most k model sizes[4] and m local edits, which is in practice much smaller than the 2^k models that are available when there are k variables in total.

[4] Usually this is a lot less, as for example in the figure 5 the interesting region seems to be somewhere between 2 and 5 variables. This reduction is even larger when there are a lot more variables (compared to the number of observations) as the solutions tend to be sparse.

	age	sex	bmi	map	tc	ldl	hdl	tch	ltg	glu	y
1	59	2	32.10	101.00	157	93.20	38.00	4.00	4.86	87	151
2	48	1	21.60	87.00	183	103.20	70.00	3.00	3.89	69	75
3	72	2	30.50	93.00	156	93.60	41.00	4.00	4.67	85	141
4	24	1	25.30	84.00	198	131.40	40.00	5.00	4.89	89	206
5	50	1	23.00	101.00	192	125.40	52.00	4.00	4.29	80	135
6	23	1	22.60	89.00	139	64.80	61.00	2.00	4.19	68	97

Table 1: Top 5 rows from the diabetes data set.

With this workflow, shinyproj allows for a more personalised accuracy/interpretability-tradeoff. This might be a great benefit, as what is interpretable can be highly subjective. For an extreme example Freitas (2014) contrasts how one user has found a decision system with 41 rules 'overwhelming' while another user has apparently analysed a set of 29,050 rules and later identified 220 of them to be interesting.

The rest of this section is organised roughly as the suggested workflow. Throughout the sections, the functionality of shinyproj is demonstrated with an application to model selection for the diabetes data set[5]. First, section 4.1 briefly discusses the fitting of the full model. After that, section 4.2 discusses the global model size selection diagnostics and 4.3 the corresponding diagnostics for local model editing. Finally, section 4.4 briefly goes through the current limitations along with suggestions for future work.

4.1 Fitting the full model

Throughout the sections, the functionality of shinyproj is demonstrated with an application to model selection in a regression problem with the diabetes data set. Table 1 shows the first five rows of the data set for a quick overview and the variables are described in section A. Before fitting the model, the data set is normalised to have 0 mean and standard deviation of 1 for numerical reasons.

The model selection process using the projection predictive approach starts with

[5]From the paper by Efron et al. (2004).

specifying the reference model which, as discussed in section 2.1.3, should represent the best prior knowledge related to the problem. Several of the variables seem quite related to each other (for example tch is a function of tc and hdl) so it is not likely that all of them are needed in the prediction.

The the linear regression is equipped with a horseshoe prior, which has, in addition to being natural for communicating the assumption about sparsity, also shown to perform well in these situations (e.g (Carvalho et al., 2009; Datta et al., 2013; Peltola et al., 2014; Van Der Pas et al., 2014)). The full model is fitted using rstanarm (Stan Development Team, 2016), which allows the hyperpriors to be specified using the latest recommendations specified in Piironen and Vehtari (2017b). The hyperprior for the effective number of non-zero is set to 3, which means that there should effectively be around 3 completely unshrunk variables.

With the projection method, specifying the reference model gives a natural set of candidate models for model selection, ie. the powerset of all the variables used in the full models. Once the full model is fit, it is enough simply to run the preliminary variable selection using the projpred package, after which we are ready to start the interpretable model selection using the functionality in shinyproj.

4.2 Choosing the model size

The interpretable variable selection starts by examining the results of the variable selection algorithm. The idea is that the user sees the global tradeoff in model size and accuracy along with a certainty metric of the decision process. With this information the user is able to make an informed choice of approximate model size which serves as a reference point for further modifications.

The upper plot shows the leave-one-out cross-validated difference in mean squared error (MSE) to the full model as a function of the model size. As can be seen, there is essentially no benefit in adding additional variables to the sub-model after 5 variables.

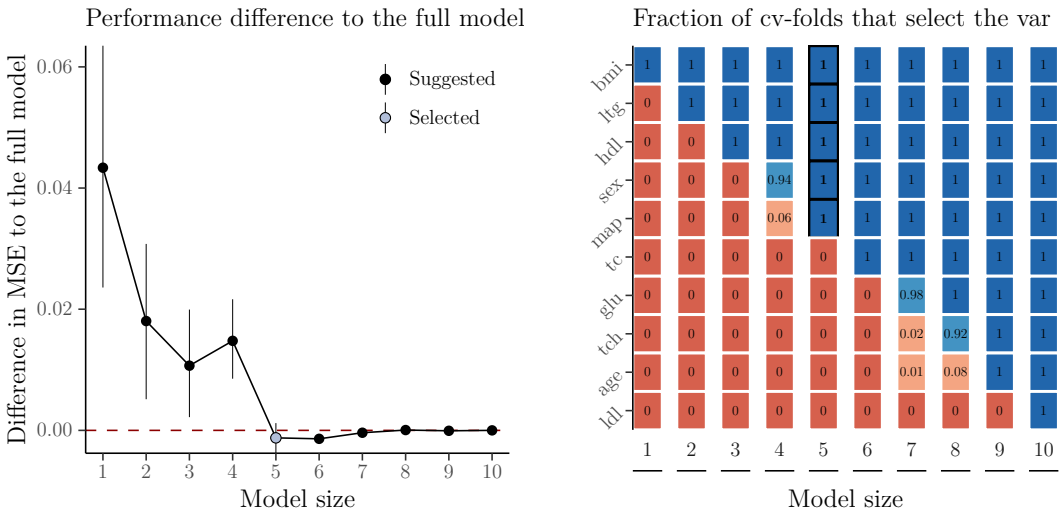

Figure 6: Plots for the model size selection.

On the other hand, as discussed in section 3.3, there might still be information to be gained. For example, even though ldl is the last variable added to the model, it does not mean that its (marginal) effect on the output is zero. A user interested in the effect of the variable could learn said effect by adding the variable to the model. Furthermore, a user preferring a simpler explanation might be satisfied with a model with only bmi, ltg and hdl as that is already very close to the full model in terms of performance.

The right-hand side plot shows that the algorithm is pretty confident about the variable order, meaning that a user mainly interested in the performance can confidently use the given variable order. This is likely because in this data set there are a lot of observations compared to the number of predictors, i.e. enough information to determine the variable order. For an example of a case where the algorithm is not so certain of the variable order and therefore a user intervention might be more beneficial, see figure B1.

4.3 Examining and editing the selected model

After the desired accuracy level / model size is selected, the user can make smaller local edits to the model after examining the model. For example, the user might have domain knowledge about the predictors which allows them to replace an uninteresting predictor in the model with a more interesting one.

On the other hand, the local edits ideally also allow the user to find models that the algorithm will not find. This is because for the algorithm, each predictor is equally costly to add to the model and therefore the selection is done only based on the predictive abilities of the predictors, whereas for the user there might be major differences between the costs (e.g. financial or interpretability) of using different predictors. Furthermore, as forward selection is path-dependent, the errors made by the algorithm might accumulate if a suboptimal choice is made early on.

The communication between the user and the model is facilitated by providing several visual diagnostics that are explained next. The first set is intended to provide information about the relation between all the predictors. This is communicated to the user based on the idea that if two predictors contain similar information, they will also likely yield similar predictions. To visualise this, the package obtains predictions for a single predictors model using each predictor as the only predictor at a time. Next, these predictions are clustered hierarchically to obtain a structure of which single predictors models are most similar. The correlations between these predictions are shown in two ways, as presented in figure 7.

The left-hand side plot projects these predictions onto 2d-plane via (metric) multidimensional scaling (Gower, 1966). In addition, the clustering-structure is drawn over the plot, with the clusters that contain the selected predictors (in bold) highlighted. This allows the user to get a quick overview of the information in the unselected parameters, which is likely also projected onto the model selected via the chosen parameters.

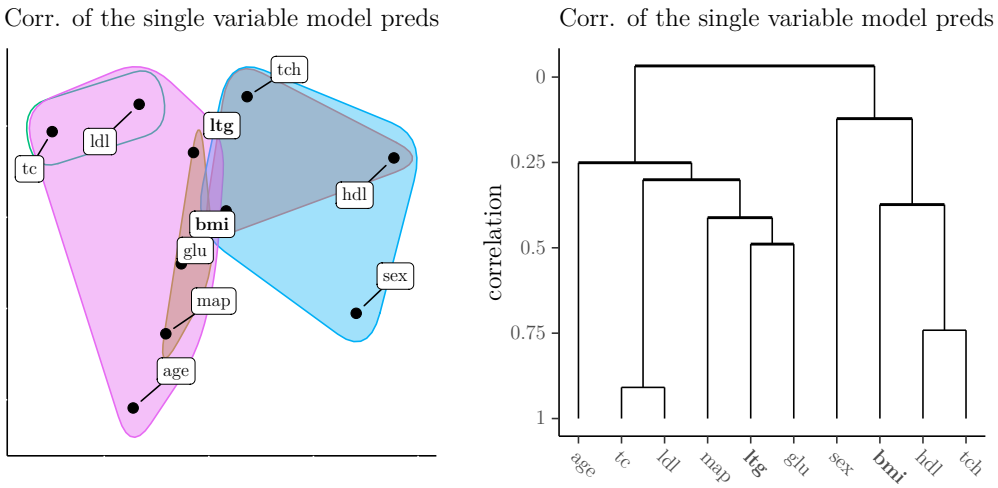

Figure 7: Plots for the correlations between single variable predictions using each of the predictors separately.

The right-hand side plot provides similar information in a different format. It is constructed from the same similarity-of-the-predictions information, but highlights the actual values of the correlations between the predictions. The y-axis in the plot can be interpreted as follows: The forks, or the vertical lines connecting different branches, constitute the lower limits of the correlations between the predictions of all the predictors in the given fork. So, as an example, the correlation between the predictions of ltg and glu is around 0.5 and and the correlation between the predictions of all the predictors from age to glu is at least 0.25.

In addition to providing information about the similarity of the predictors, the plots can also be used to infer the dissimilarity of the predictors. For example, in the figure the predictors are are roughly in two big clusters (coloured pink and and light blue). Therefore, in order to use the information in the predictors as efficiently as possible, one would likely want to select predictors from each cluster. In practice this is also what happens in this case as the best two-predictors model according to the algorithm uses the predictors bmi and ltg.

In addition to these plots, shinyproj contains a histogram of the coefficients and a pairs plot for the predictors included in the selected model. The plots are shown in

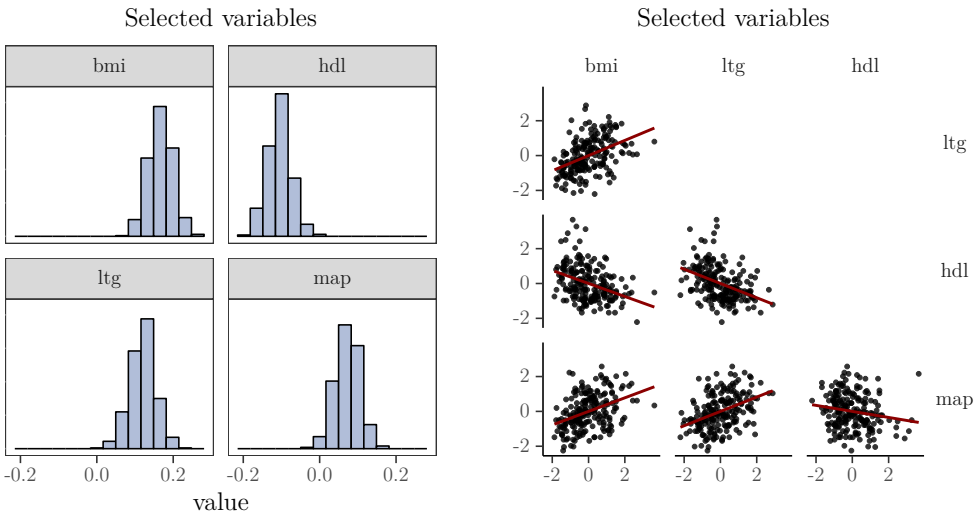

Figure 8: Plots for examining the selected model.

figure 8. The plot on the left shows the effects of the selected predictors on the target while the plot on the right shows the correlation structure within the predictors, as this is quite often also important in understanding the model.

By adding or removing predictors to or from the selected model, the user sees how the coefficients of the predictors change, which is important especially in trying to interpret how the selected model works in relation to the entire data. An example of this is illustrated in figure B2.

Another way to examine how well the model fits to the data is to compare samples from the posterior distribution of the model to the observed data. Figure 9 presents 100 samples from the posterior predictive distribution of the best model with 5 variables (as suggested by the variable selection algorithm). In some sense these posterior predictive checks can be regarded similarly as unit tests. They might reveal problems in the model, but seeing that the posterior predictive distribution matches with the data does not confirm the validity of the model.

Furthermore, the plot above shows only one type of a posterior predictive check, which is also the only one implemented in shinyproj currently. To be more thorough, the user would often also like to compare other summaries as well, for example obtain

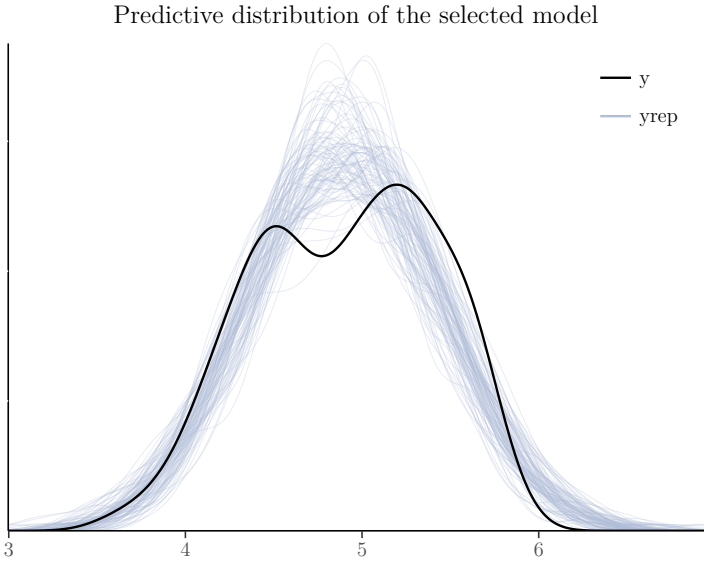

Figure 9: Posterior predictive distribution of the selected model.

draws of certain quantiles from the model and compare these to the corresponding summaries of the data. A more thorough treatment of posterior predictive checks can be found from e.g. Gelman et al. (1996).

This shows that even though the model performs really well, it does not accurately capture for example the bimodality in the data, suggesting that adding more variables might be preferable (if it fixed the problem, which the user could try by simply adding variables that could explain the bimodality and seeing how the posterior predictive distribution changes). In some situations this could be a concern: For example, if the two modes corresponded to two relatively different subgroups, giving the same predictions for both of the subgroups might be problematic. In this case, however, the data does not really allow to separate the two groups as even the full model yields a qualitatively similar posterior predictive distribution.

Last, as the user performs the local edits, it is useful to keep the models suggested by the algorithm as reference points. This way, the user sees how much of the accuracy of the model they trade to improve the interpretability of the model, allowing for efficient tradeoffs. As an example, the user could replace hdl (the amount of 'bad cholesterol') in the model with tch (total cholesterol divided by hdl), if this measure

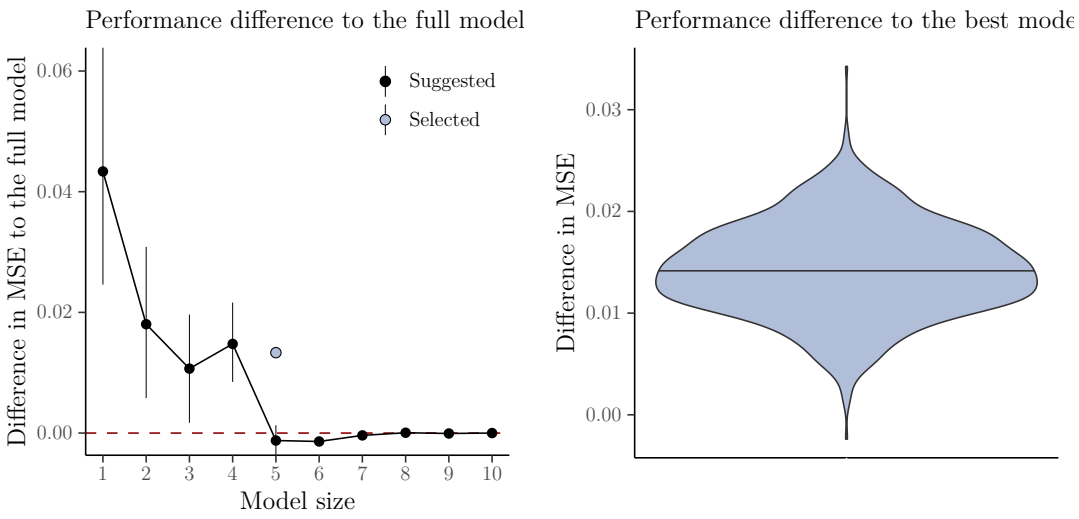

Figure 10: Plots showing the performance difference compared to the best submodel.

is more interpretable to them.

Figure 10 presents the plots that show this to the user. The right-hand side plot shows the bootstrapped difference to the 'best model', in this case the model containing hdl instead of tc/hdl. As is shown, it is likely that the loss in MSE is likely to be between 0.01 and 0.02. Now, depending on the context the increased interpretability might justify the change, or if the user might decide to include both of the variables to the model, if adding more variables is not a problem.

In addition, the mean of this difference is added to the plot showing the performance difference of the 'best' sub-models of all sizes, so that the user also a reference of different model sizes available.

4.4 Limitations

One of the problems in the current version of shinyproj is that it does not really scale for datasets with a lot of variables. For example, it is not possible to fit a pairs plot with more than 10 variables to the screen. Similarly, the plots containing information about the correlation between the single-variable models (presented in figure 7) will

become cluttered when there are around 50 variables.

In some sense this is not a problem as typically models with more than 20 variables or so would be really hard, if not impossible to interpret anyway. Therefore, it would possible to simply hide all the variables that the model selection algorithm selects after a certain limit, as they would not be considered relevant for the model anyway. For interpretability's sake this is, however, not an option, not at least as done naively by simply not showing some of the variables, as this would result in the user being erroneously confident about being considered all the possible options when building the model.

A possible approach might be to combine the interface with user modelling, which could interactively assist the user by learning the user's preferences. For example, the model could learn the variables that the user is likely to want to interpret with respect to the model and show only those (and correspondingly hide those that are uninteresting to the user).

Another case not considered in this thesis is when the reference model is of different class than the submodels. For example, the projection could be done from a Gaussian process or a hierarchical model to a linear one, which could provide multiple benefits. First of all, the user sees the lost accuracy when restricting the set of models to linear ones. In addition, a more advanced reference model would contain more information to be projected onto the submodels, which would furthermore also improve the projection. As the linear submodels would retain the interpretability this might prove to be a fruitful direction for further research in interpretable model selection.

5 Conclusions

Interpretability in model selection is a significant issue that concerns the users of models in many ways. While there exist current tools for model selection, they do not always bear interpretability in mind; if they do, tradeoffs between interpretability and other issues related to building and using the model are not always efficient. Interpretability in models is useful for both the users of the models and for those whom the decisions of the models concern.

This thesis has discussed Bayesian predictive variable selection with a focus on interpretable models. Section 2 reviewed the most common methods for model selection, along with variable selection heuristics. At the end of the section, I presented some problems that demonstrate the need for occasional user intervention.

Section 3 has discussed the issue of interpretable models in detail. In the context of models, interpretabily closely related to the accessibility of the information contained in the model. Thus, interpretability is not only the transparency or the usability of the model, but combines both traits along with others. The properties that make models interpretable are reviewed along with examples of interpretable and non-interpretable models. Especially linearity and sparsity are found to be in key roles in guaranteeing interpretability of the model.

There are multiple benefits to be gained from interpretable models, ranging from facilitating debugging and user intervention to fairness and public trust in algorithmic modelling and its results. In trust, interpretability is key, as it enables the users to trust the model and the predictions it produces. This is of vital importance especially when the models are used to aid high-stakes decision-making, such as in health-care or the criminal justice system. At the end of the section, the idea of interpretability is extended to the entire model selection process. This means that in addition to the model itself, the user should be able to understand and interpret the decisions made throughout the model building process and how the build of the model and its results is affected by those decisions.

Finally, Section 4 has combined these ideas and introduced shinyproj, a new tool for interpretable variable selection. The functionality of the tool has been demonstrated with an application to an example data set. While the tool definitely streamlines the process of interpretable model selection, its limitations are acknowledged. Furthermore, as discussed at the end of the section, while shinyproj does have its limitations, it will likely prove a fruitful platform for further improvements.

References

H. Akaike. A new look at the statistical model identification. *IEEE transactions on automatic control*, 19(6):716–723, 1974.

J. Angwin, J. Larson, S. Mattu, and L. Kirchner. Machine bias. *Pro Publica*, 2016.

J. M. Bernardo. Expected information as expected utility. *The Annals of Statistics*, pages 686–690, 1979.

M. Betancourt. A unified treatment of predictive model comparison. *arXiv preprint arXiv:1506.02273*, 2015.

T. Calders and S. Verwer. Three naive bayes approaches for discrimination-free classification. *Data Mining and Knowledge Discovery*, 21(2):277–292, 2010.

T. Calders and I. Žliobaitė. Why unbiased computational processes can lead to discriminative decision procedures. In *Discrimination and Privacy in the Information Society*, pages 43–57. Springer, 2013.

R. Caruana, H. Kangarloo, J. Dionisio, U. Sinha, and D. Johnson. Case-based explanation of non-case-based learning methods. In *Proceedings of the AMIA Symposium*, page 212. American Medical Informatics Association, 1999.

C. M. Carvalho, N. G. Polson, and J. G. Scott. Handling sparsity via the horseshoe. In *International Conference on Artificial Intelligence and Statistics*, pages 73–80, 2009.

A. Chouldechova. Fair prediction with disparate impact: A study of bias in recidivism prediction instruments. In *3rd Workshop on Fairness, Accountability, and Transparency in Machine Learning*, 2016.

J. Datta, J. K. Ghosh, et al. Asymptotic properties of bayes risk for the horseshoe prior. *Bayesian Analysis*, 8(1):111–132, 2013.

W. Dieterich, C. Mendoza, and T. Brennan. Compas risk scales: Demonstrating accuracy equity and predictive parity. *Northpoint Inc*, 2016.

J. A. Dupuis and C. P. Robert. Variable selection in qualitative models via an entropic explanatory power. *Journal of Statistical Planning and Inference*, 111(1): 77–94, 2003.

D. K. Duvenaud, H. Nickisch, and C. E. Rasmussen. Additive gaussian processes. In *Advances in neural information processing systems*, pages 226–234, 2011.

B. Efron, T. Hastie, I. Johnstone, R. Tibshirani, et al. Least angle regression. *The Annals of statistics*, 32(2):407–499, 2004.

A. Eklund, T. E. Nichols, and H. Knutsson. Cluster failure: why fmri inferences for spatial extent have inflated false-positive rates. *Proceedings of the National Academy of Sciences*, page 201602413, 2016.

T. Elomaa. In defense of c4. 5: Notes on learning one-level decision trees. In *Proc. 11th Int. Conf. on Machine Learning*, pages 62–69, 1994.

M. Forina et al. An extendible package for data exploration, classification and correlation. *Institute of Pharmaceutical and Food Analisys and Technologies, Via Brigata Salerno*, 16147, 1991.

A. A. Freitas. Comprehensible classification models: a position paper. *ACM SIGKDD explorations newsletter*, 15(1):1–10, 2014.

O. H. Gandy. Engaging rational discrimination: exploring reasons for placing regulatory constraints on decision support systems. *Ethics and Information Technology*, 12(1):29–42, 2010.

A. E. Gelfand, D. K. Dey, and H. Chang. Model determination using predictive distributions with implementation via sampling-based methods. Technical report, STANFORD UNIV CA DEPT OF STATISTICS, 1992.

A. Gelman, X.-L. Meng, and H. Stern. Posterior predictive assessment of model fitness via realized discrepancies. *Statistica sinica*, pages 733–760, 1996.

A. Gelman, A. Jakulin, M. G. Pittau, and Y.-S. Su. A weakly informative default prior distribution for logistic and other regression models. *The Annals of Applied Statistics*, pages 1360–1383, 2008.

I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. *3rd International Conference on Learning Representations (ICLR2015)*, 2015.

B. Goodman and S. Flaxman. European union regulations on algorithmic decision-making and a "right to explanation". *arXiv preprint arXiv:1606.08813*, 2016.

C. Goutis and C. P. Robert. Model choice in generalised linear models: A bayesian approach via kullback-leibler projections. *Biometrika*, 85(1):29–37, 1998.

J. C. Gower. Some distance properties of latent root and vector methods used in multivariate analysis. *Biometrika*, 53(3-4):325–338, 1966.

B. Kim. *Interactive and interpretable machine learning models for human machine collaboration*. PhD thesis, Massachusetts Institute of Technology, 2015.

S. Kullback and R. A. Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.

B. Letham, C. Rudin, T. H. McCormick, D. Madigan, et al. Interpretable classifiers using rules and bayesian analysis: Building a better stroke prediction model. *The Annals of Applied Statistics*, 9(3):1350–1371, 2015.

M. Lichman. UCI machine learning repository, 2013. URL `http://archive.ics.uci.edu/ml`.

Z. C. Lipton. The mythos of model interpretability. In *ICML Workshop on Human Interpretability of Machine Learning*, 2016.

J. R. Lloyd, D. K. Duvenaud, R. B. Grosse, J. B. Tenenbaum, and Z. Ghahramani. Automatic construction and natural-language description of nonparametric regression models. 2014.

Y. Lou, R. Caruana, and J. Gehrke. Intelligible models for classification and regression. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 150–158. ACM, 2012.

Y. Lou, R. Caruana, J. Gehrke, and G. Hooker. Accurate intelligible models with pairwise interactions. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 623–631. ACM, 2013.

J. E. Oakley and A. O'Hagan. Shelf: the sheffield elicitation framework (version 3.0). *School of Mathematics and Statistics, University of Sheffield, UK*, 2016.

T. Peltola, A. S. Havulinna, V. Salomaa, and A. Vehtari. Hierarchical bayesian survival analysis and projective covariate selection in cardiovascular event risk prediction. In *Proceedings of the Eleventh UAI Conference on Bayesian Modeling Applications Workshop-Volume 1218*, pages 79–88. CEUR-WS. org, 2014.

J. Piironen and A. Vehtari. Comparison of bayesian predictive methods for model selection. *Statistics and Computing*, 27(3):711–735, 2017a.

J. Piironen and A. Vehtari. Sparsity information and regularization in the horseshoe and other shrinkage priors. *arXiv preprint arXiv:1707.01694*, 2017b.

R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2017. URL `https://www.R-project.org/`.

A. E. Raftery and Y. Zheng. Discussion: Performance of bayesian model averaging. *Journal of the American Statistical Association*, 98(464):931–938, 2003.

M. T. Ribeiro, S. Singh, and C. Guestrin. Model-agnostic interpretability of machine learning. In *ICML Workshop on Human Interpretability of Machine Learning*, 2016a.

M. T. Ribeiro, S. Singh, and C. Guestrin. Why should i trust you?: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1135–1144. ACM, 2016b.

D. J. Spiegelhalter, N. G. Best, B. P. Carlin, and A. Van Der Linde. Bayesian measures of model complexity and fit. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 64(4):583–639, 2002.

Stan Development Team. rstanarm: Bayesian applied regression modeling via Stan., 2016. URL `http://mc-stan.org/`. R package version 2.13.1.

R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.

S. Van Der Pas, B. Kleijn, A. Van Der Vaart, et al. The horseshoe estimator: Posterior concentration around nearly black vectors. *Electronic Journal of Statistics*, 8(2): 2585–2618, 2014.

M. Veale. Logics and practices of transparency and opacity in real-world applications of public sector machine learning. In *4th Workshop on Fairness, Accountability, and Transparency in Machine Learning*, 2017.

A. Vehtari and J. Lampinen. Model selection via predictive explanatory power. *Report B38, Laboratory of Computational Engineering, Helsinki University of Technology*, 2004.

A. Vehtari, J. Ojanen, et al. A survey of bayesian predictive methods for model assessment, selection and comparison. *Statistics Surveys*, 6:142–228, 2012.

A. Vehtari, A. Gelman, and J. Gabry. Practical bayesian model evaluation using leave-one-out cross-validation and waic. *Statistics and Computing*, 27(5):1413–1432, 2017.

S. Watanabe. Asymptotic equivalence of bayes cross validation and widely applicable information criterion in singular learning theory. *Journal of Machine Learning Research*, 11(Dec):3571–3594, 2010.

Y. Yao, A. Vehtari, D. Simpson, A. Gelman, et al. Using stacking to average bayesian predictive distributions. *Bayesian Analysis*, 2018.

M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(1):49–67, 2006.

R. E. Zapata-Vázquez, A. O'Hagan, and L. Soares Bastos. Eliciting expert judgements about a set of proportions. *Journal of Applied Statistics*, 41(9): 1919–1933, 2014.

J. Zeng, B. Ustun, and C. Rudin. Interpretable classification models for recidivism prediction. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 180(3):689–722, 2017.

Z. Zhang and D. B. Neill. Identifying significant predictive bias in classifiers. In *4th Workshop on Fairness, Accountability, and Transparency in Machine Learning*, 2017.

I. Žliobaitė. Measuring discrimination in algorithmic decision making. *Data Mining and Knowledge Discovery*, pages 1–30, 2017.

A Description of the diabetes data set

The diabetes data set contains 442 observations of the following variables:

- age: the age of the patient.
- sex: the sex of the patient, 1 for female and 2 for male.
- bmi: the Body mass index of the patient.
- map: the mean arterial pressure (ie. blood pressure) of the patient.

The rest of the variables are not well described in the paper, but appear to be as follows

- tc: total cholesterol level of the patient.
- ldl: low density lipoprotein (ie. the 'bad cholesterol') level of the patient.
- hdl: high density lipoprotein (ie. the 'good cholesterol') level of the patient.
- tch: tc divided with hdl, appears to be rounded for some patients.
- ltg: possibly a glucose-related measurement?
- glu: a measurement related to the glucose level of the patient.
- y: a target measure related to diabetes risk of the patient.

B Additional figures

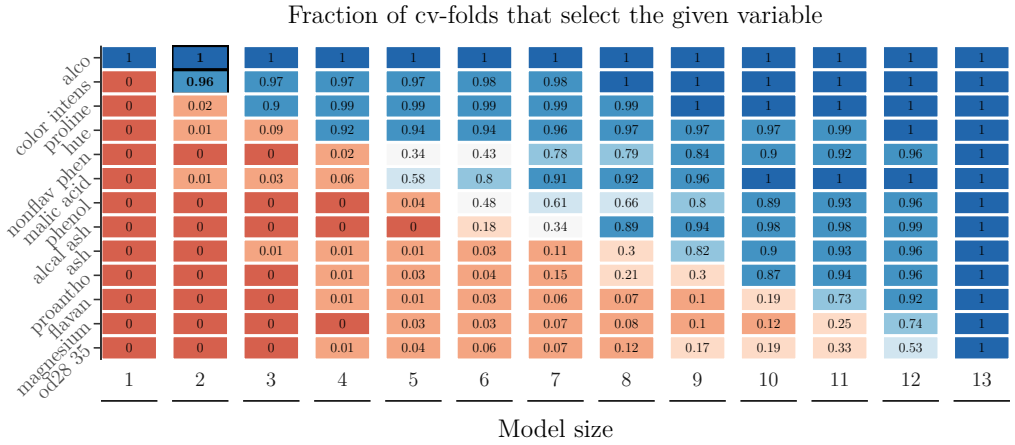

Figure B1: An example with the wine data set (Forina et al., 1991; Lichman, 2013) where there is much less certainty in variable order. This means that there is less certainty in the suggested variable combinations for any model combinations and therefore more input from the user might be beneficial.

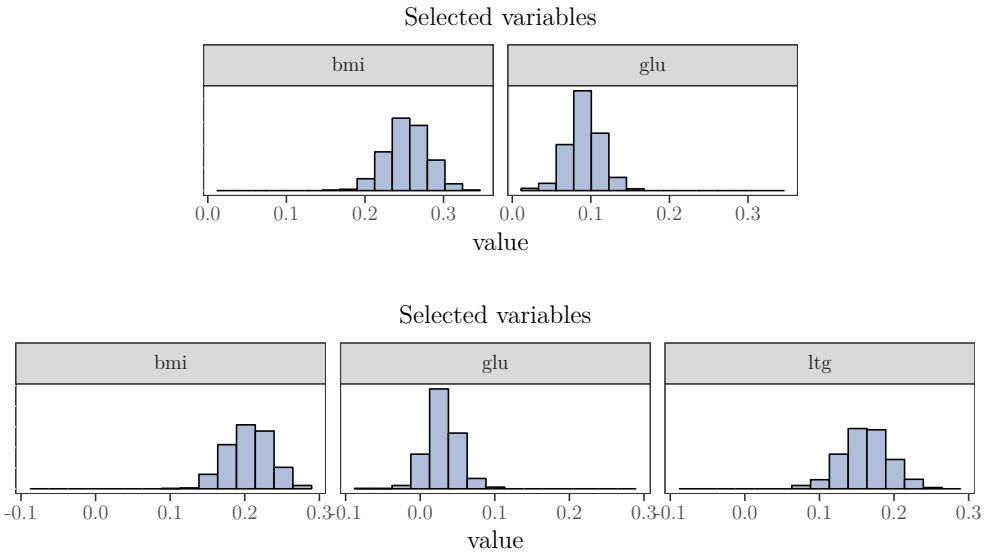

Figure B2: An example with the diabetes data set. Notice how adding ltg to the model reduces the estimated effect of glu as before the information related to ltg was likely projected onto glu.